Body Paint

Real-time interactive virtual painting created from motion capture dance performance

CIS 499 SENIOR PROJECT DESIGN DOCUMENT

Cassandra Ichniowski
Advisor: Alla Safonova
University of Pennsylvania

PROJECT ABSTRACT

Generally, in dance performance, choreographers, lighting designers, or set designers create a setting for a piece: physical constructions, lights and light cues, painted backdrops, pre-created computer backdrops. With new technology, however, backdrops, scenes, and ambiance can be created from a performance instead of imposed upon a performance. Over the last couple decades, many dancers, graphics artists, and computer scientists have explored some of the potential applications of computer graphics and animation to the creation, education and performance of dance. Computers can interpret feedback from the dance performance and create graphical displays or lighting controls based on the qualities of the movement.

Body Paint is a real-time interactive motion-capture program where a dancer will get immediate "feedback" on her movements in the form of a 2-dimensional virtual "painting." The dancer has a variety of options: the quality of the stroke (oil paints, charcoals, or ink, etc.), the view of the "painting," and the way the mapping is generated. Other factors, such as color and weight of the stroke, will be interpreted by the algorithm and based solely on her movement. When performing with Body Paint, not only will the dancer's movement influence the developing image, but the image will in turn influence how the dancer continues moving and uses the performance space.

Project blog:
http://cassandrai.wordpress.com

1. INTRODUCTION

Body Paint will interactively create a virtual painting based on the physical properties (path, velocity, acceleration, and angular velocity) of the movements of the dancer. The default process will do a "conventional" painting – movements will map to a "canvas" at the front of the space. Ideally, there will be two other mappings: a painted "forest" where the motions will be traced and marked in place, with strokes applied, and a moveable cross-section will be projected; and a footwork painting, which only paints from the motion of the footsteps on the ground. These strokes will be selected at the beginning of the piece from a small group: oil paintbrush, ink/calligraphy, and charcoal/pastel. These may also be changed in the resulting painting after the fact. The color of the medium will be interpreted based solely on these properties, although the spectrum can be limited to the dancer's specifications.

1.1. Significance of Problem or Production/Development Need

Body Paint is an artistic tool helping choreographers and dancers intelligently create an ambiance or setting for a piece, rather than imposing an atmosphere. It can be used in performance, choreographed or improvised; or, the dancer can create a painted backdrop to be used for future performances elsewhere.

1.2. Technology

Vicon Motion Systems, ViconBlade, C++, OpenGL, GLUT, projection system.

1.3. Design Goals

1.3.1 Target Audience.

Dance performances generally include some sort of performance environment, with lighting, scrims, painted flats, and occasionally set pieces. Body Paint is a tool for choreographers or solo dancers to create an interesting 2D backdrop for their piece. It can be created during the performance (in theory), recorded to play later, or used as a planning tool for creating the atmosphere (eg, projections and lighting).

1.3.2 User goals and objectives

Users will suit up in the MoCap suit and perform a pre-choreographed or improvised piece. They have many options of stroke to apply to their painting and can redraw it in any other stroke. They also have the option of randomly varying color throughout the painting or maintaining one color throughout. They can alter the staging or choreography of their piece based on the feedback from the painting or simply allow the algorithm to interpret the pre-choreographed piece. Dancers can also use the resulting painting to make design decisions for performance, such as backdrop or lighting.

1.3.3 Project features and functionality

The positions of the dancer's end effectors are streamed into a parser frame by frame. Each marker position is then drawn to the screen using various brushes including Pencil, Calligraphy, Chalk, Ink, Airbrush, or Pointillism. Users can choose to have

2. RELATED WORK

As early as 1985 in the San Francisco Ballet's "Pixellage," computer graphics have been used to generate backdrops for dance performances. Using an Aurora 100 videographics workstation, Darryl Sapien created animated backdrops relating to the pieces choreographed by Betsy Erikson. Some complimented pieces, but others provided props and interacted with the live dancers. [1]

One very extensive performance-graphics collaboration resulted in DigitalBeing. DigitalBeing is an ambient intelligent environment using pressure and physiological sensors to control lighting and projected light imagery to project the dancer's arousal state. [2]

Brushstroke:
DAB: Interactive Haptic Painting with 3D Virtual Brushes
Bill Baxter Vincent Scheib Ming C. Lin Dinesh Manocha

- a deformable, 3D brush model; force feedback
- This paper develops a number of formulae based on physical properties of real painting, eg translucency, blending, wetness, and drying. They solves the problem of creating realistic brushstrokes by modeling brushes in 3D and applying various physical equations to the resulting stroke. It is considerably more complex than my current needs.

Artistic Reality: Fast Brush Stroke Stylization for Augmented Reality
Jan Fischer Dirk Bartz Wolfgang Straßer

- Pointilistic result on video/real images

Painterly Rendering with Curved Brush Strokes of Multiple Sizes
Aaron Hertzmann

- creating an image with a hand-painted appearance from a photograph
- brush sizes for expressing various levels of detail
- "We model brush strokes as anti-aliased cubic B-splines, each with a given color and thickness. Each stroke is rendered by dragging a circular brush mask along the sweep of the spline."

Sketching with a Low-latency Electronic Ink Drawing Tablet
Alex Henzen Neculai Ailenei Fabian DiFiore Frank Van Reeth John Patterson

- Sketching tablet to mimic real-life drawing/sketching
- Smooths curve a la clean-up artist
- Instead, a high-level internal representation, using cubic Bezier curves, is created, using the classical formula

Essentially, few contemporary papers concern themselves with applying 2D strokes to paths. Generally, they are more concerned with painterly rendering from images/video or 3D brushes. However, these have proven useful in: 1) showing me the potential complexity involved in creating realistic brushstrokes and 2) helping devise a method of abstracting from the data, specifically using Bezier curves.

3. PROJECT DEVELOPMENT APPROACH

3.1. Algorithm Details

Rendering Brushstrokes:

i. Pencil
ii. Calligraphic
iii. Ink – tapering, solid color
iv. Chalk – patchy
v. Airbrush – fade to transparent
vi. Pointilistic – random pixels in radius, with complementary color

Color Variance:
Randomly select new R, G, and B values
Set as Target color
Interpolate between current color and target color until target color is reached
Repeat from top

RealTime:
Although RealTime streaming never completed successfully, I did learn the process to connect to the server and request data.

1) Capture Range Of Motion starting and ending in T-Pose
2) Process ROM and label markers for skeleton
3) Configure the appropriate skeleton and calibrate character
4) Turn on the RTE (check box in RealTime window), change to solving data, and replay last
5) Connect to RTE on port 801 using ExampleClient cpp code in SDK
6) Send an ERequest for data using ExampleClient (type and packet)
7) Position of markers and joint angles are stored in markerPositions and bodyPositions in ExampleClient and available for use
8) Loop from 5 until finished streaming
9) Disconnect

Range of Motion should should the full range of motion of each joint in the skeleton.

Parsing:
The following ASCII txt file is of a C3D file from a 41-marker capture. It looks like this:

```
# Pts =        330
# Video Frames =             817

Video Frame #                1
 1           -362.6837      43.40683      1576.704
 2           -391.7587     -107.3856      1560.726
 3           -488.6526      53.48709      1578.006
 4           -519.5791     -71.79614      1573.069
 5           -554.4143      20.64053      1427.9
 6           -596.244       49.57842      1269.839
 7           -430.8454     -20.91483      1346.435
 8           -374.684      -60.96157      1146.681
 9           -628.6105     -39.1553       1339.783
 10          -446.8915      154.9754      1386.962
 11          -442.9143      271.4127      1194.888
 12          -451.0745      311.3223      1064.325
```

13	-462.3891	317.7682	948.5731
14	-409.9991	204.2796	881.577
15	-461.8405	342.2488	861.0051
16	-486.0468	267.6412	780.2259
17	-555.4429	-138.7922	1392.447
18	-627.4448	-210.8626	1261.953
19	-687.7206	-238.7719	1075.776
20	-709.5269	-250.2236	991.2256
21	-620.7932	-220.8743	870.3996
22	-771.7914	-247.8921	870.6739
23	-700.681	-180.0732	778.1686
24	-393.1987	63.97879	905.2348
25	-517.5219	-151.8896	920.0466
26	-595.6268	76.04768	1002.06
27	-635.605	3.702953	1003.569
28	-463.9663	119.9345	595.1468
29	-488.104	73.03046	476.7895
30	-555.2372	74.33335	265.7897
31	-570.1176	-8.914515	59.72726
32	-598.7126	-75.56767	38.81243
33	-387.37	-0.2057196	35.0409
34	-441.6114	52.32135	32.16083
35	-689.7778	-102.2426	634.0278
36	-675.8574	-60.07012	485.6354
37	-773.78	14.40037	405.8848
38	-864.5709	190.9764	180.6218
39	-831.2443	247.275	221.4229
40	-855.8621	232.1203	16.32042
41	-918.2637	220.6685	36.0695
42	-543.6483	2.262916	927.4525
43	-492.767	79.68206	819.6555
44	-586.0266	-73.37332	819.9297
45	-406.9134	-27.15499	480.9724
46	-592.1981	-41.48679	457.6575
47	-581.0893	-56.57289	68.84749
48	-830.1472	221.2171	184.1876
49	-462.5262	-26.05782	42.9954
50	-851.9534	243.0234	66.03599
51	-403.8961	-14.26323	23.52061
52	-862.3765	251.8694	6.377307
53	-521.7049	3.977246	1047.661
54	-516.4247	11.93174	1169.653
55	-516.8362	13.71464	1291.713
56	-539.7396	17.1433	1391.145
57	-550.7799	12.82319	1488.176
58	-510.596	-3.22294	1580.064
59	-434.8227	190.0163	1310.845
60	-572.4491	-159.1584	1323.188
61	-493.727	299.6649	1040.53
62	-711.3784	-185.4905	1049.787
63	-478.1609	276.2814	902.2861
64	-711.7213	-186.7248	906.6749
65	-470.4121	264.5554	833.0958
66	-711.8584	-187.342	835.0844
67	-487.3497	264.7611	795.5177
68	-712.7498	-168.1415	794.5577
69	-504.3559	263.2525	766.9912
70	-714.327	-148.8724	764.0426
71	-500.1729	245.2178	801.2093
72	-703.6982	-145.9238	804.9122
73	-634.0964	57.19005	860.8679
74	-595.764	73.64761	967.9793
75	-630.942	15.84041	968.1165
76	-456.4232	65.07597	970.5165
77	-559.3516	-103.8884	970.7908

78	-392.2387	71.52185	880.8913
79	-535.831	-164.1642	881.3027
80	-488.7898	44.84687	813.2781
81	-556.4715	-66.31028	813.4839
82	-493.7956	-28.25216	816.9811
83	-472.9493	35.24662	862.2394
84	-540.6311	-75.84196	862.4451
85	-481.041	74.67622	816.364
86	-488.8583	69.87609	800.1121
87	-501.0644	80.36779	809.5066
88	-493.3156	85.16792	825.6899
89	-469.315	79.54491	786.2603
90	-482.961	72.34473	785.0945
91	-489.3383	85.23649	779.403
92	-475.4866	106.837	791.8148
93	-407.1877	-17.1433	525.065
94	-420.8337	-24.41206	523.899
95	-427.211	-11.45172	518.2076
96	-413.565	-4.251538	519.3734
97	-390.1815	-31.88654	486.7326
98	-416.5822	-54.03568	487.0069
99	-447.1658	-9.120235	496.6071
100	-398.4103	-3.565806	475.6923
101	-578.4835	-80.50494	811.9067
102	-583.7637	-68.91606	798.3978
103	-593.7753	-65.00739	813.4839
104	-583.2836	-71.52185	827.4042
105	-589.3867	-88.11656	782.2145
106	-589.2495	-72.75616	783.5174
107	-604.6785	-72.61902	783.7917
108	-612.3586	-96.89393	796.4091
109	-583.8322	-53.07566	500.3101
110	-583.6265	-37.71526	501.6129
111	-599.0555	-37.57811	501.8872
112	-599.2612	-52.93851	500.5844
113	-579.4435	-54.44712	456.2861
114	-571.4205	-21.18912	459.029
115	-617.1588	-18.24047	489.3383
116	-610.3701	-59.31582	456.3546
117	-379.5526	-41.55536	460.6747
118	-437.2227	-61.51016	496.2643
119	-422.3423	-20.22909	487.0069
120	-407.3934	20.9834	477.7495
121	-430.1597	-38.19527	394.0216
122	-446.2058	-45.39546	401.3589
123	-453.2689	-27.84072	403.0733
124	-434.2741	-12.41175	393.953
125	-534.7338	-55.81858	146.7466
126	-550.7114	-63.01877	154.084
127	-557.843	-45.46403	155.7983
128	-537.3396	-25.92067	145.718
129	-525.1335	-70.90469	95.04246
130	-589.0438	-96.61964	94.56244
131	-582.9408	-44.98402	117.603
132	-566.2089	-15.29182	59.59011
133	-580.6093	-50.94989	423.851
134	-568.3347	-2.948647	473.978
135	-600.3583	-34.14945	471.7836
136	-632.3135	-65.41883	469.5893
137	-633.6849	5.485856	392.3073
138	-630.1877	20.36624	403.6219
139	-645.891	16.45757	413.565
140	-655.7655	-4.594404	401.839
141	-776.4543	163.1356	228.2116
142	-772.9571	178.016	239.5262

143	-788.6603	174.1759	249.4693
144	-801.7579	149.901	237.5376
145	-785.986	176.576	176.096
146	-788.6603	241.9263	197.2851
147	-816.5011	204.0738	229.1716
148	-862.1022	189.9478	181.9247
149	-453.886	-51.77277	14.19465
150	-469.4521	-9.120235	-6.788747
151	-591.3067	-76.66483	5.965868
152	-583.4894	-97.9911	16.52614
153	-586.7809	-47.93267	90.44805
154	-557.2944	-37.44097	89.96804
155	-849.2105	281.3558	62.88162
156	-896.1832	269.6984	69.39608
157	-856.7535	285.4016	199.068
158	-833.233	291.2304	195.8451
159	-825.347	204.691	188.9877
160	-828.9814	202.4281	158.747
161	-410.8906	-32.57227	-8.434504
162	-480.8353	-12.34318	-4.868697
163	-465.2692	-54.9957	16.18328
164	-870.1939	287.0474	13.02891
165	-895.0174	269.4241	77.35057
166	-848.0447	281.0815	70.90469
167	-472.4008	-3.085794	1341.772
168	-597.4097	48.3441	1374.893
169	-513.339	93.25955	1374.893
170	-568.4033	-42.37824	1378.047
171	-424.7424	-25.78352	1196.671
172	-628.4048	56.64146	1190.911
173	-460.1947	146.5409	1190.979
174	-570.3233	-124.8718	1197.219
175	-431.5311	79.68206	1083.457
176	-504.9045	-101.214	1087.64
177	-485.9783	161.4899	1385.521
178	-596.1068	-109.9228	1391.83
179	-458.2747	-15.70326	1591.31
180	-482.001	27.56643	1615.379
181	-522.4592	-4.662978	1633.756
182	-498.7329	-48.00124	1609.687
183	-491.6013	-1.92005	1506.416
184	-508.6074	71.59042	1532.748
185	-595.3525	15.84041	1574.989
186	-552.7	-62.12732	1531.72
187	-465.9549	-6.377307	1489.479
188	-518.6191	7.200186	1463.215
189	-477.0638	7.337332	1552.086
190	-484.4011	36.61809	1538.097
191	-488.6526	10.90314	1538.371
192	-483.7839	-22.90345	1549.823
193	-502.2987	-44.02399	1532.062
194	-495.3728	-19.26907	1536.108
195	-536.5167	29.07504	1013.512
196	-553.1115	-3.42866	1016.461
197	-562.6431	-2.948647	968.5279
198	-546.0484	29.55505	965.5792
199	-608.6557	46.42406	1027.981
200	-618.1188	46.97264	980.0482
201	-527.2593	33.5323	1131.732
202	-542.0026	0.0685732	1134.612
203	-543.8541	-3.22294	1085.857
204	-529.1108	30.17221	1082.977
205	-601.7984	46.49263	1133.721
206	-603.6499	43.06397	1084.897
207	-549.6827	34.83519	1451.009

208	-559.3516	5.760149	1450.117
209	-554.4143	4.662978	1430.3
210	-544.7455	33.73801	1431.191
211	-610.7815	39.36102	1435.511
212	-605.8442	38.33242	1415.694
213	-532.0594	22.42344	1503.947
214	-539.2596	-5.897295	1502.576
215	-545.4312	-3.42866	1484.198
216	-538.231	24.89207	1485.57
217	-589.3867	20.9834	1523.011
218	-595.5582	23.45203	1504.633
219	-445.3829	196.3936	1315.714
220	-446.2744	186.3134	1305.291
221	-434.5484	177.7417	1305.771
222	-433.5883	187.822	1316.263
223	-436.1941	217.2399	1279.576
224	-439.8971	205.7882	1275.736
225	-428.0339	203.1824	1272.101
226	-422.8224	205.5139	1294.868
227	-491.8755	295.7562	1071.319
228	-495.5785	284.3045	1067.479
229	-483.7153	281.6987	1063.845
230	-480.081	293.1504	1067.685
231	-497.9786	306.8651	1050.953
232	-515.739	284.1673	1045.604
233	-485.2925	284.8531	1049.101
234	-479.601	304.8764	1046.153
235	-584.2437	-154.9069	1328.811
236	-577.3177	-147.5695	1317.634
237	-562.7117	-151.2725	1317.497
238	-569.6376	-158.6784	1328.674
239	-596.5182	-176.096	1291.988
240	-590.0038	-165.6043	1287.667
241	-580.335	-173.353	1283.485
242	-577.6606	-178.976	1306.937
243	-705.5496	-184.2562	1081.399
244	-699.0352	-173.7645	1077.079
245	-689.3664	-181.5133	1072.896
246	-695.8808	-191.9364	1077.216
247	-719.4014	-186.7934	1060.896
248	-713.2984	-157.6498	1055.067
249	-693.6179	-182.3361	1057.81
250	-705.9611	-200.0966	1055.41
251	-499.7615	310.088	1046.77
252	-482.2067	305.2879	1036.552
253	-486.8011	287.8017	1036.827
254	-509.7732	289.5161	1036.141
255	-492.1498	298.0877	997.8086
256	-483.5096	295.8248	999.1801
257	-485.9783	287.2531	1000.346
258	-498.1843	287.3903	998.9058
259	-478.1609	277.0357	873.3483
260	-469.5207	274.7728	874.7197
261	-471.9893	266.2012	875.8855
262	-480.561	268.4641	874.514
263	-477.0638	276.0757	834.7416
264	-459.8519	271.6184	837.4159
265	-464.7206	254.4751	839.816
266	-481.8639	258.9324	837.0731
267	-721.9387	-187.2734	1058.427
268	-709.184	-198.2451	1045.604
269	-698.6923	-183.3647	1043.273
270	-714.6699	-166.4957	1045.33
271	-717.8243	-186.9305	1006.86
272	-710.4183	-192.2107	1006.929

273	-705.1382	-184.7362	1006.792
274	-713.2984	-175.2731	1006.792
275	-718.1671	-188.0963	878.0112
276	-710.6926	-193.3078	878.0798
277	-705.4125	-185.9019	878.0112
278	-712.887	-180.6218	877.9427
279	-724.5444	-189.5363	839.4045
280	-709.6641	-200.028	839.4731
281	-699.1724	-185.1476	839.3359
282	-714.0527	-174.6559	839.2674
283	-472.1265	270.7956	824.867
284	-469.4521	264.0068	823.6327
285	-476.378	260.7153	826.7185
286	-479.0524	267.5041	827.9528
287	-486.664	277.7215	799.7692
288	-480.6981	262.5668	797.0263
289	-487.624	259.3438	800.1121
290	-493.5899	274.4985	802.9236
291	-716.5214	-188.4391	824.7299
292	-708.5669	-186.9991	825.2099
293	-710.2812	-179.1132	829.05
294	-718.1671	-180.4847	828.5014
295	-725.0244	-176.4388	795.8605
296	-707.4011	-173.2845	796.9577
297	-709.1155	-165.33	800.7978
298	-726.7388	-168.4843	799.6321
299	-491.2584	278.1329	793.2548
300	-485.2925	262.8411	790.5118
301	-490.4355	260.2353	793.7348
302	-496.4014	275.5271	796.4777
303	-503.1216	277.1043	773.3
304	-500.5844	261.5382	764.7969
305	-505.7959	258.9324	768.0198
306	-508.3331	274.4985	776.5229
307	-726.4645	-171.7759	788.6603
308	-708.704	-168.6215	789.8261
309	-710.0069	-162.5871	793.6662
310	-727.7673	-165.81	792.5005
311	-727.5616	-158.2669	767.3341
312	-710.0755	-151.2039	762.3282
313	-711.3784	-145.238	766.1683
314	-728.8646	-152.3011	771.1742
315	-477.5438	263.4582	819.0383
316	-474.3208	254.818	821.3012
317	-481.1096	254.0637	828.0214
318	-484.3325	262.7725	825.7585
319	-498.4586	249.9493	796.6834
320	-495.2357	241.2405	798.9464
321	-501.9558	240.4862	805.7351
322	-505.1788	249.195	803.4722
323	-711.5841	-178.7703	820.1354
324	-702.5324	-177.1246	824.867
325	-707.264	-171.0901	831.8615
326	-716.3842	-172.7359	827.1299
327	-705.8925	-149.7639	799.0149
328	-696.7723	-148.1181	803.7465
329	-701.5724	-142.0837	810.7409
330	-710.6241	-143.7294	806.0094

Video Frame #		2	
1	-361.3808	4.44681	1576.429
2	-389.8386	-107.9342	1560.863
3	-487.4183	52.80136	1578.006
4	-518.4819	-72.61902	1572.932
5	-553.18	20.22909	1428.037

```
6           -596.1068      49.44128      1270.387
7           -430.7768     -21.18912      1346.366
8           -373.5868     -61.23587      1146.612
9           -628.062      -39.36102      1340.195
10          -445.7258      154.4954      1386.962
```

The format is “markerNumber X Y Z” within a frame. Each frame is notated with “Video Frame # frameNumber”. Pseudocode for parsing:

```
Read in the number of Markers and number of Frames
While not end of file
        Extract LFIN, RFIN, LTOE, RTOE from frame
        Tokenize line
        Add [x1,y1,x2,y2,x3,y3,x4,y4] to frame
```

3.2. Tools used

3.2.1 Hardware

Vicon Motion Capture system, projection display

3.2.2 Software

OS: Mac, OpenGL, GLUT, ViconBlade

4. WORK PLAN

4.1.1. Project Milestone Report (Alpha Version)

- Exhaustive literature review on similar Dance/MoCap projects
- Exhaustive literature review of 2D brushstroke algorithm and refinements
- Thorough brushstroke algorithm outline and pseudo-code
- Some Vicon training and familiarization

4.1.2. Project Final Deliverables

- Resources for future Vicon Blade development and troubleshooting
- Fully functional drawing application which can use mouse input to paint
- Six brushstrokes for painting: Pencil, Calligraphy, Chalk, Ink, Airbrush, and Pointillism
- Random color variability option
- Loading, parsing, and interpreting ASCII txt converted c3d files

4.1.3 Project timeline.

1. Background research
    a. Similar dance projects
    b. Brushstroke algorithms
2. Learn Vicon hardware and ViconBlade software
    a. Training
    b. Practice running the system
    c. Figure out realtime engine and needs regarding filetypes and skeletons
3. Creating brushstrokes

        a. Read similar 2D algorithms
        b. Code up and fine tune brushstroke algorithms
            i. Pencil
            ii. Calligraphic
            iii. Ink – tapering, solid color
            iv. Chalk – patchy
            v. Airbrush – fade to transparent
            vi. Pointilistic – random pixels in radius, with complementary color
4. Combine MoCap and brushstrokes
    a. Write a parser for an ASCII txt file of a c3d file
    b. Apply stroke to path created from c3d file (x,y,z)
    c. Adjust effects on color, width, canvas position
5. User trials/Results!

5. RESULTS

File Preparation
Use C3D Data program to convert *.c3d to *.txt (Program is available from here: ftp://c3dftp:c3dftp@ftp.c3d.org/user/c3d-data.exe)

Work Environment Setup (Mac OSX)
1) Open project in XCode
2) Ensure the following frameworks are included:
    a. GLUT, AGL, Carbon, OpenGL, ApplicationServices
3) Input requests path to .txt file, e.g.
    /Users/Cassandra/paintbrush/MoCap/05_10_markers.txt

Control (Mac OSX)
F1 – fullscreen
g – draw parsed motion!
m – toggle mouse drawing on/off
c – toggle color variance on/off
r – refresh screen (blank canvas)
1 – Pencil
2 – Ink
3 – Chalk
4 – Calligraphy (tends to crash program, be careful!)
5 – OilPaint alpha version
6 – Pointillism
7 – Airbrush

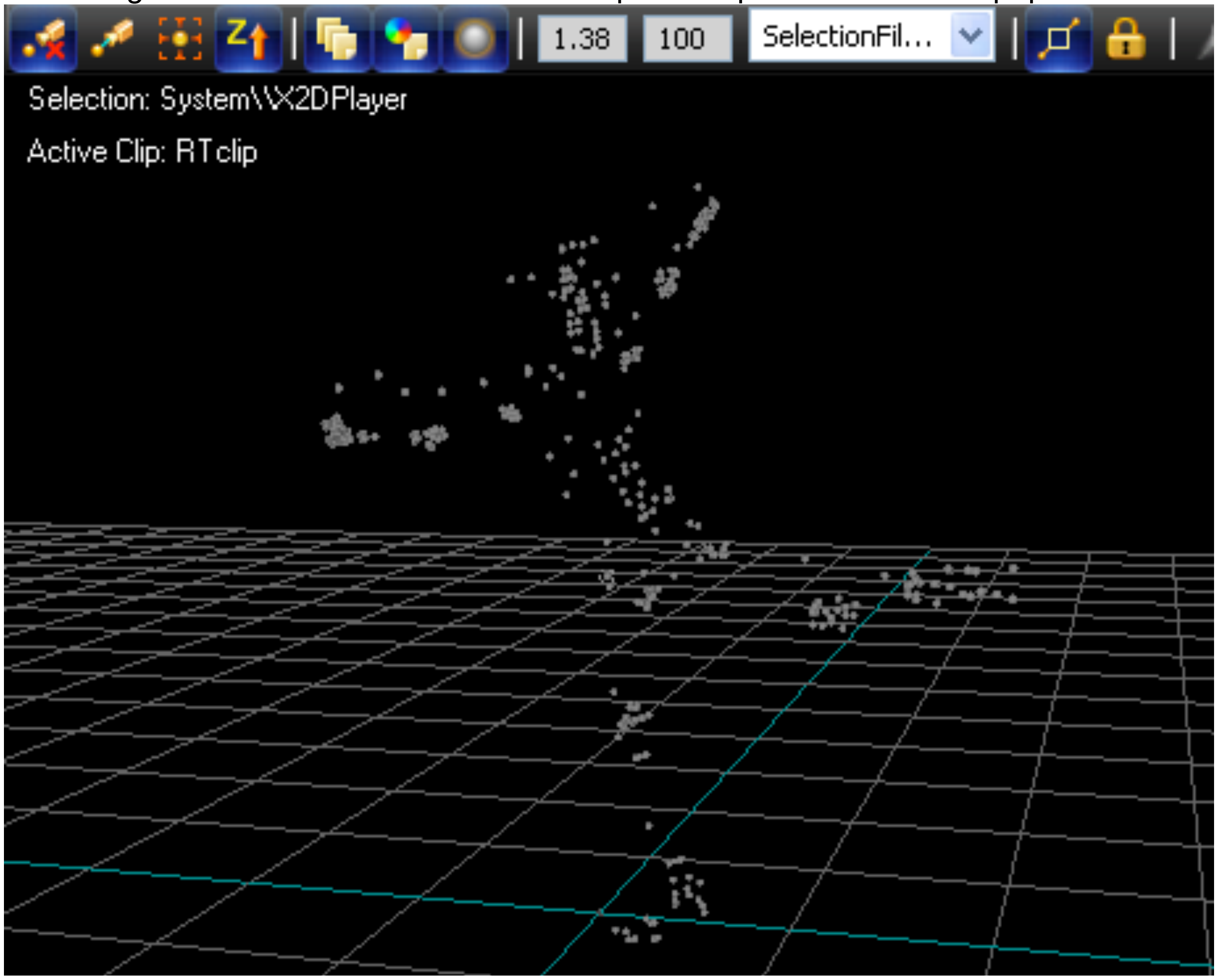
Problems:
Vicon Blade:
-Flickering Markers
-Bad automatic labeling and solving
-Errors loading skeletons

-Unable to calibrate characters
	e.g. “Constraint RIHAND_R_Wrist has no parameters. Its offset will not be altered by calibration.”

RealTime:
-*port 801*
-ExampleClient code: EInfo, ERequest yielding bad packet and type

Brushes:
-Structure

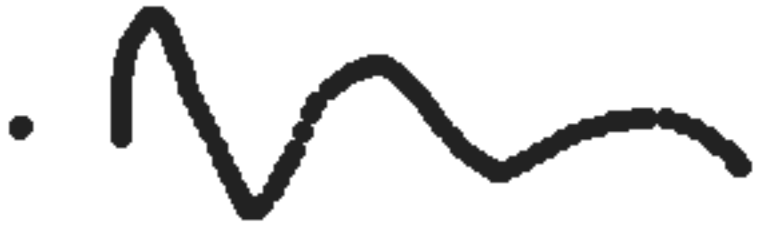
Motion Capture:
-File types:
C3D adds many points and the parser does not label them. This code only works on files using the same marker set as CMU. http://mocap.cs.cmu.edu/info.php

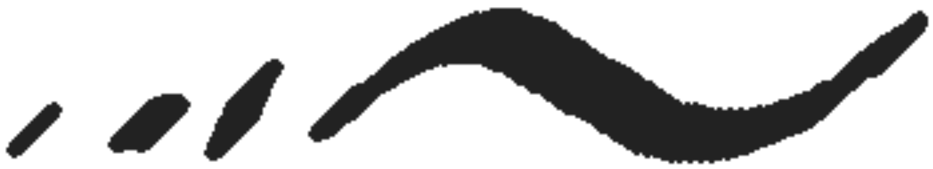


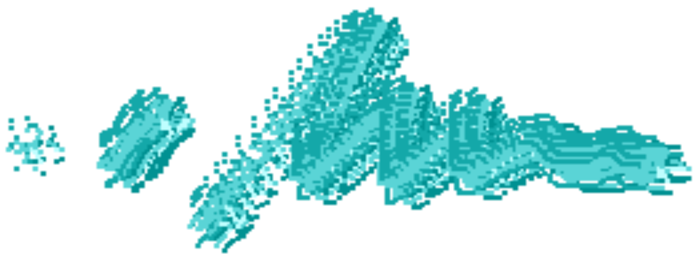
Selection: System\\X2DPlayer
Active Clip: RTclip

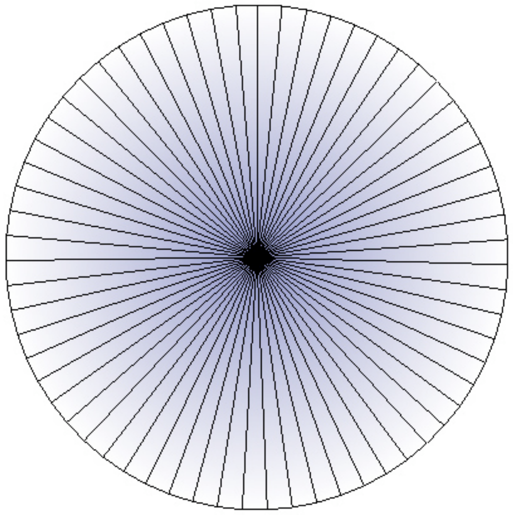
Parser:
-Structure

Results:
Brushes from Mouse Input

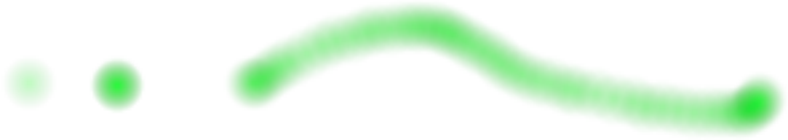

Pencil


Calligraphy

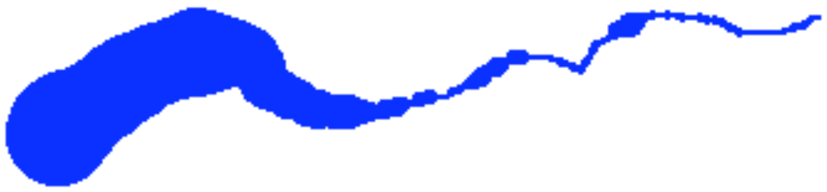

Chalk



Airbrush introduced me to the concept of transparency from a coding perspective. OpenGL has alpha channels as a built in structure to blend a newly drawn pixel with the pixels behind it. Each point is a triangle fan drawn with an alpha value of .5 in the center and 0 at the edges. See picture at left: the triangle fan is drawn over this purple Airbrush stroke shape.

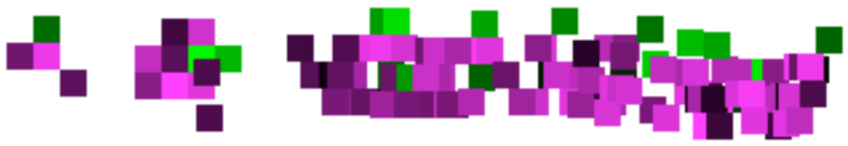


Airbrush



Ink

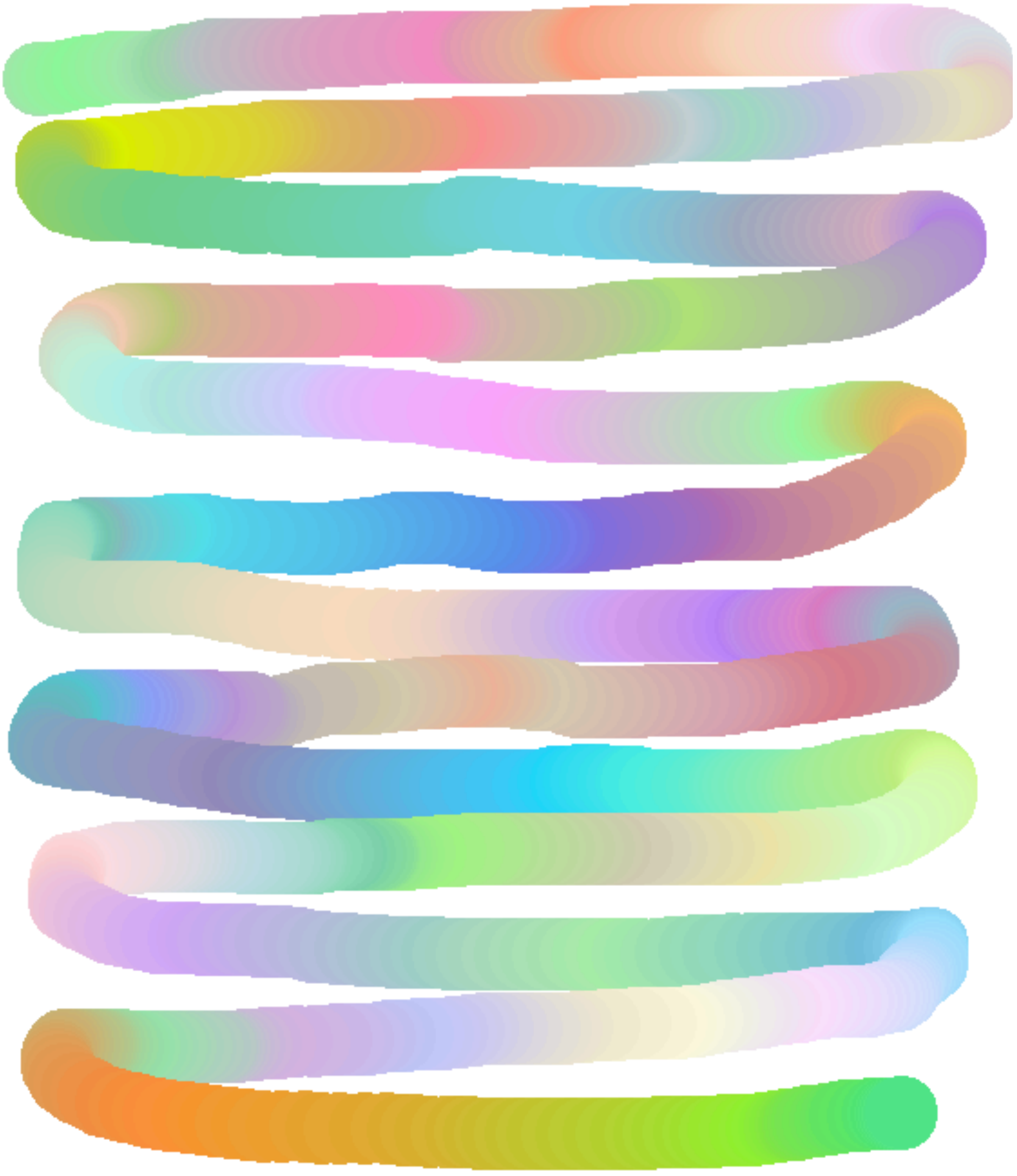
Originally, I considered Ink and Calligraphy under the same heading. I later realized Ink was more about variability in size of stroke while Calligraphy was a standard, slanted brushstroke repeated in the same size. Ink resizes dynamically with the speed of the stroke: faster movement enlarges the stroke, and slower movement shrinks the stroke. Minimum and maximum radii are preset.



Pointillism

Inspired by the seminal work of Georges Seurat and the Pointillist movement, this brush paints small squares in different values to the screen. The algorithm selects a random point within the brush radius, and draws a relatively sized square in the current color with randomly selected value (in painting, amount of black in the color). One-fifth of the time, it also paints a random square of the complementary color. Above, the color is magenta and the complementary color is green.

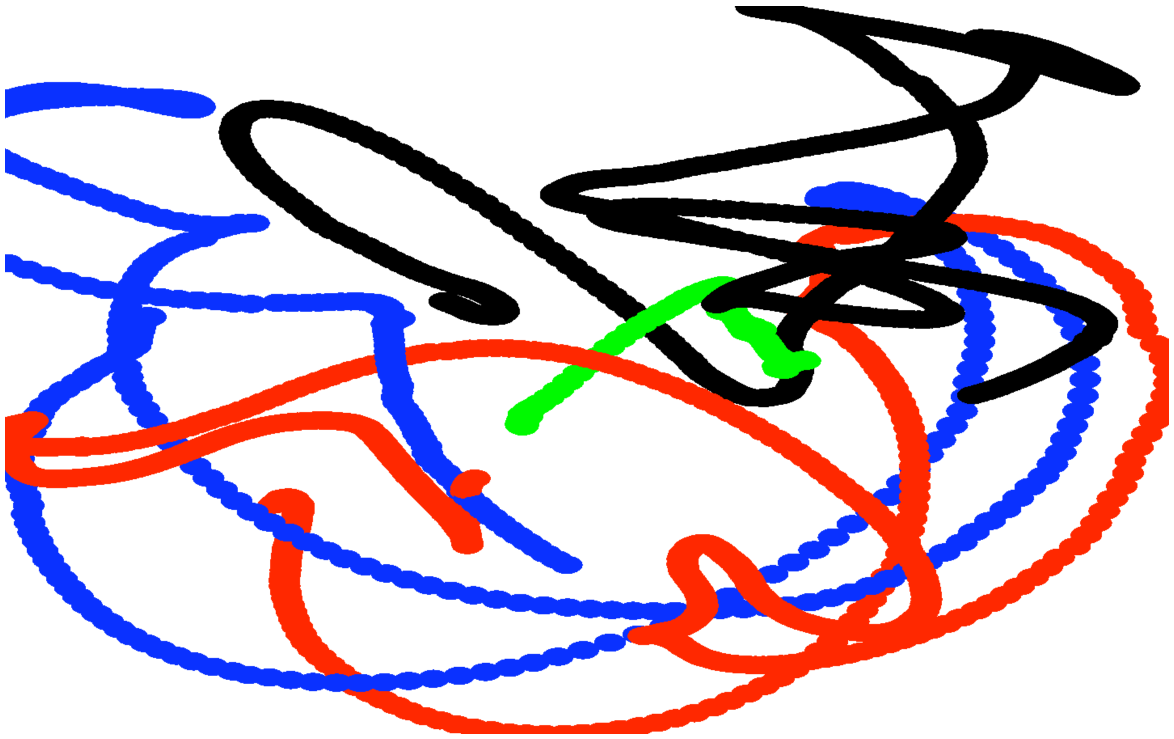
Random Color Changing



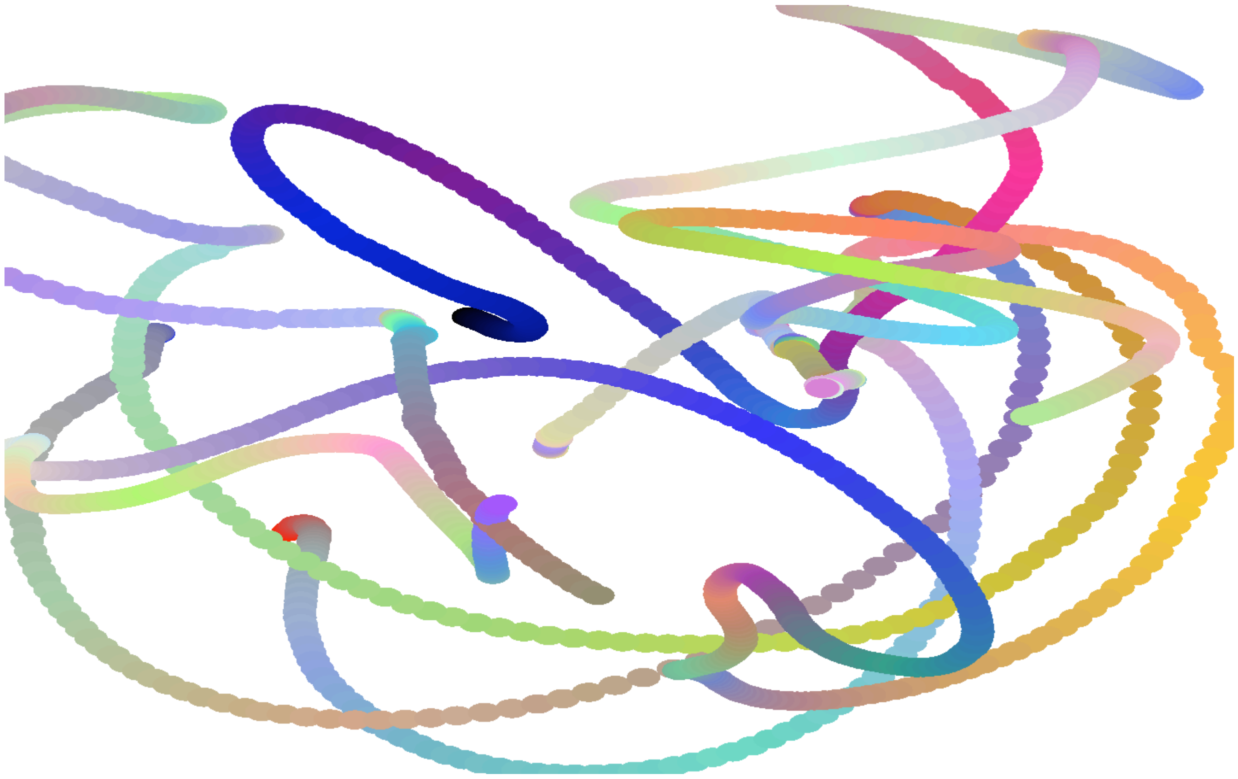
Brushes with MoCap Input
File: MoCap/05_10_markers.txt
dance - glissade devant, glissade derriere, attitude/arabesque

Left hand = black	Left toe = green
Right hand = red	Right toe = blue

Pencil stroke

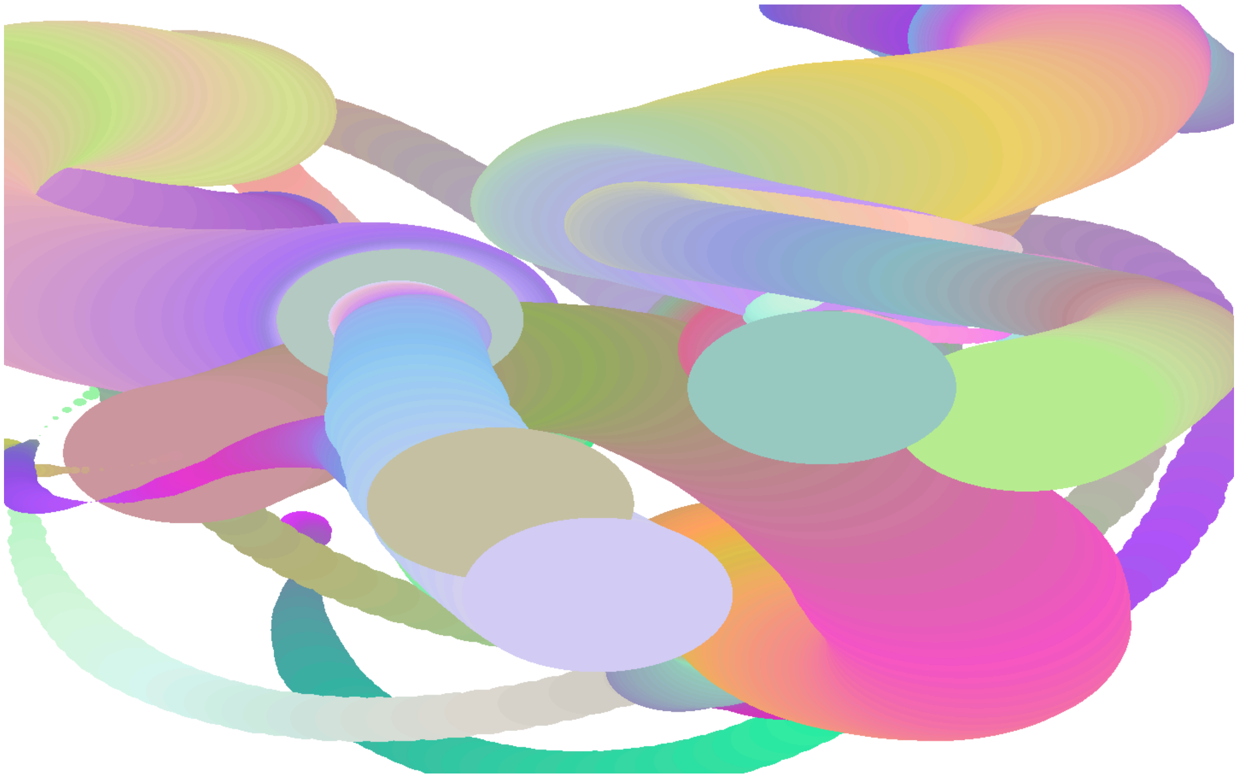


Pencil with color variance

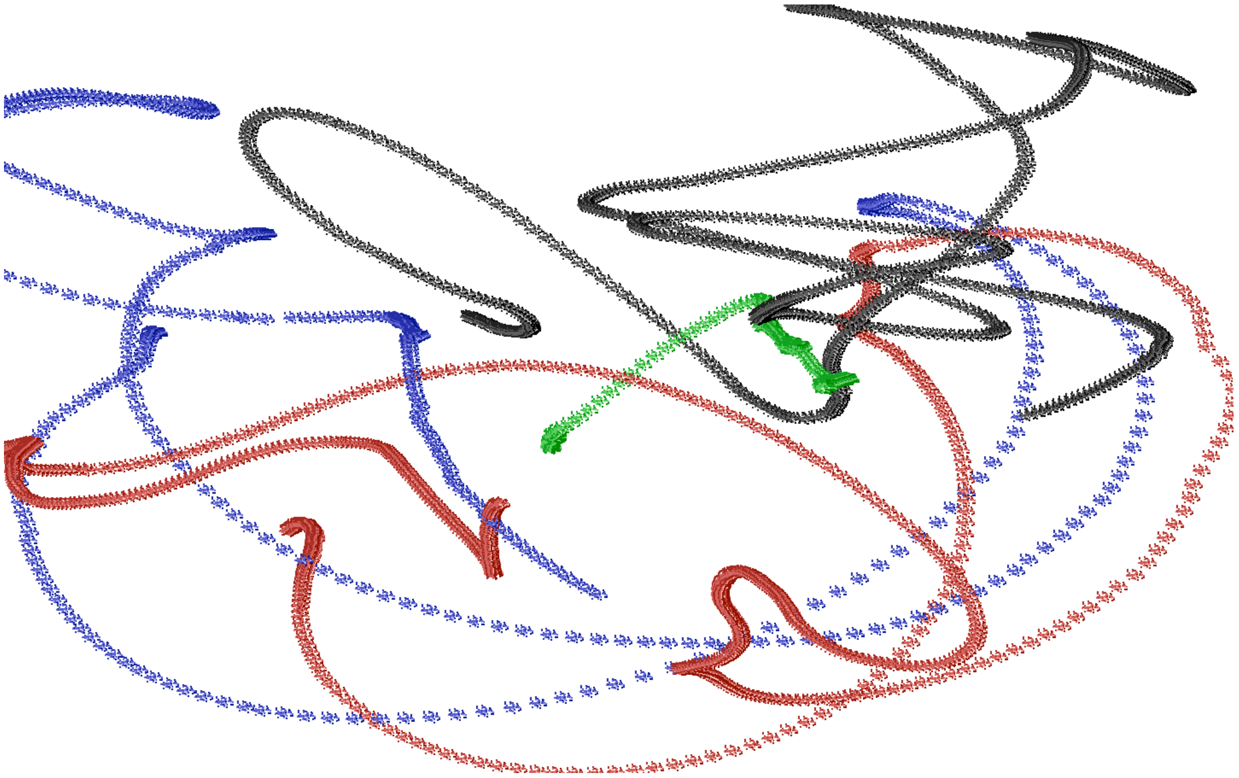


Ink

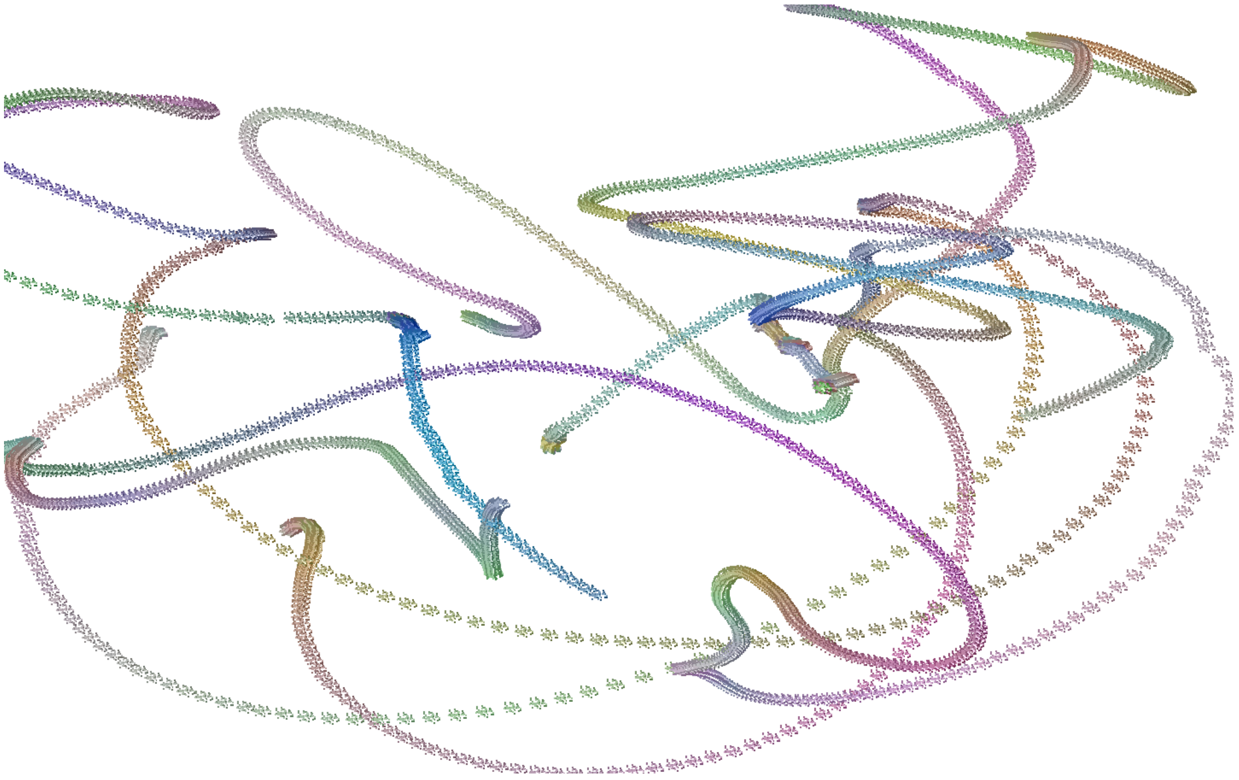
Ink with Color variance



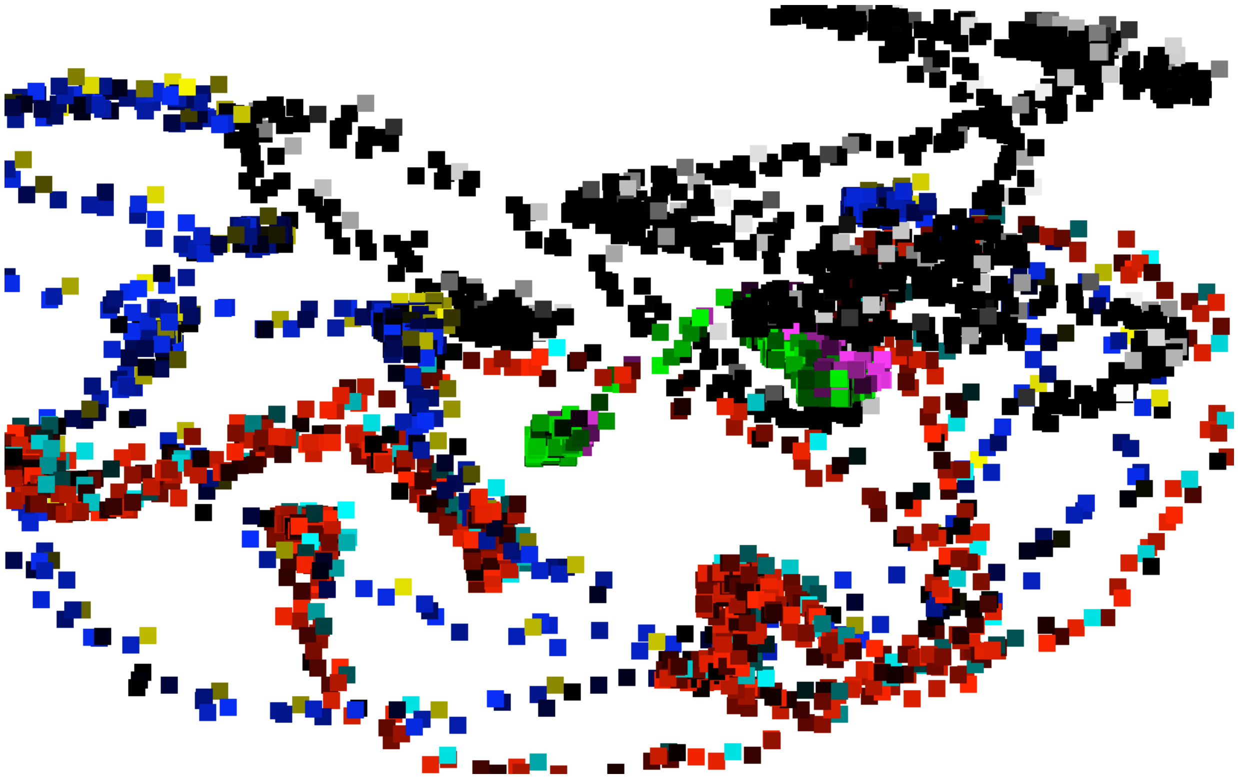
Chalk



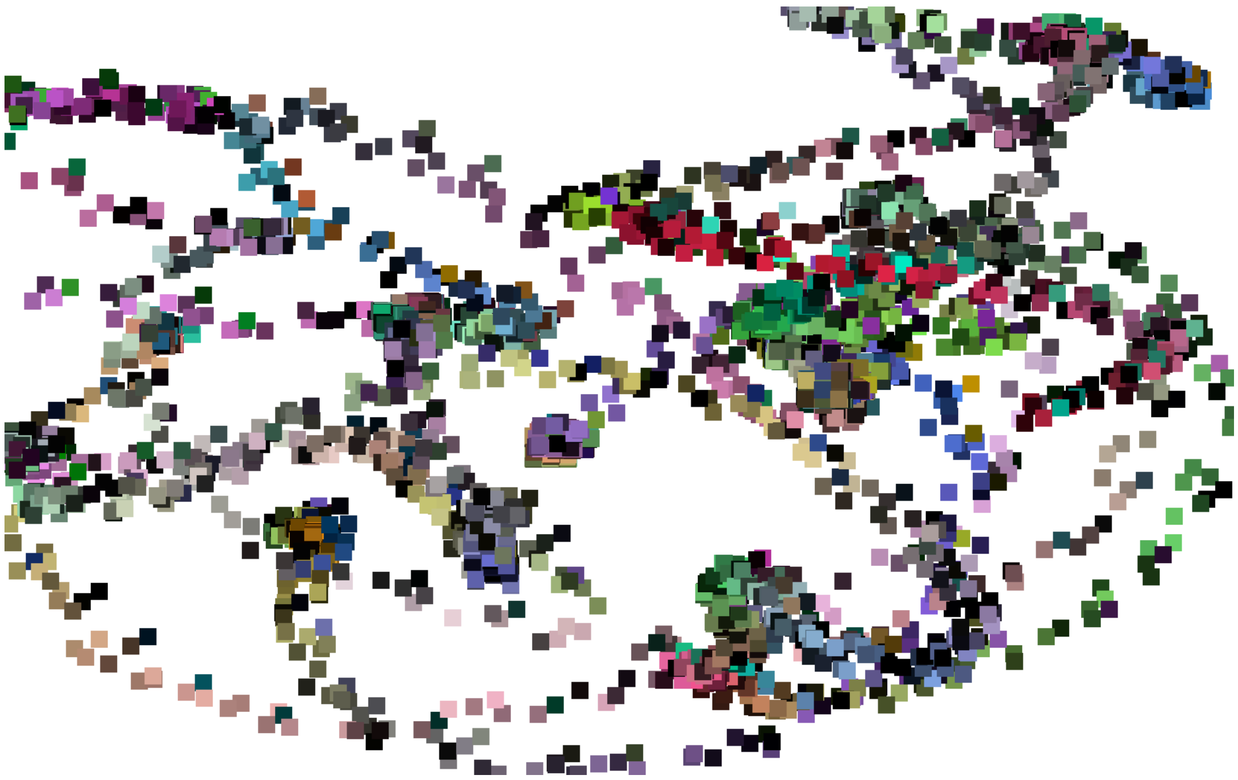
Chalk with Color Variance



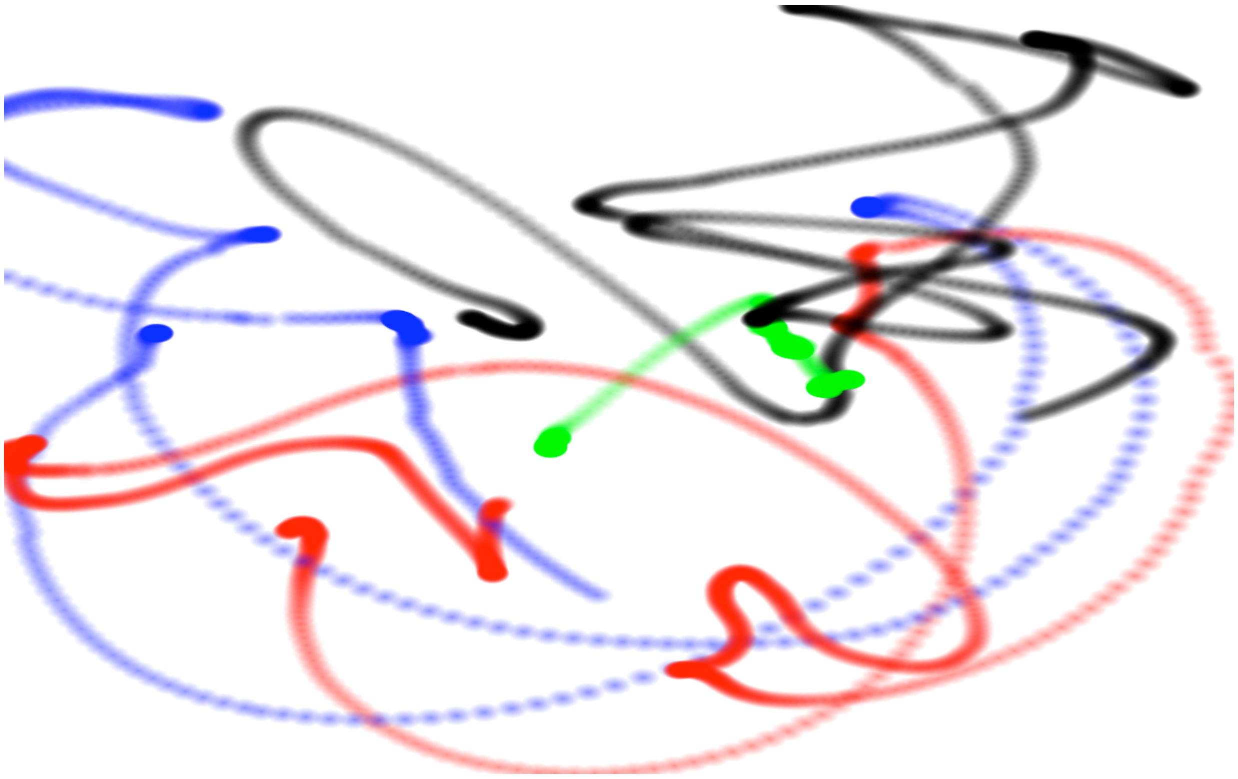
Pointillism



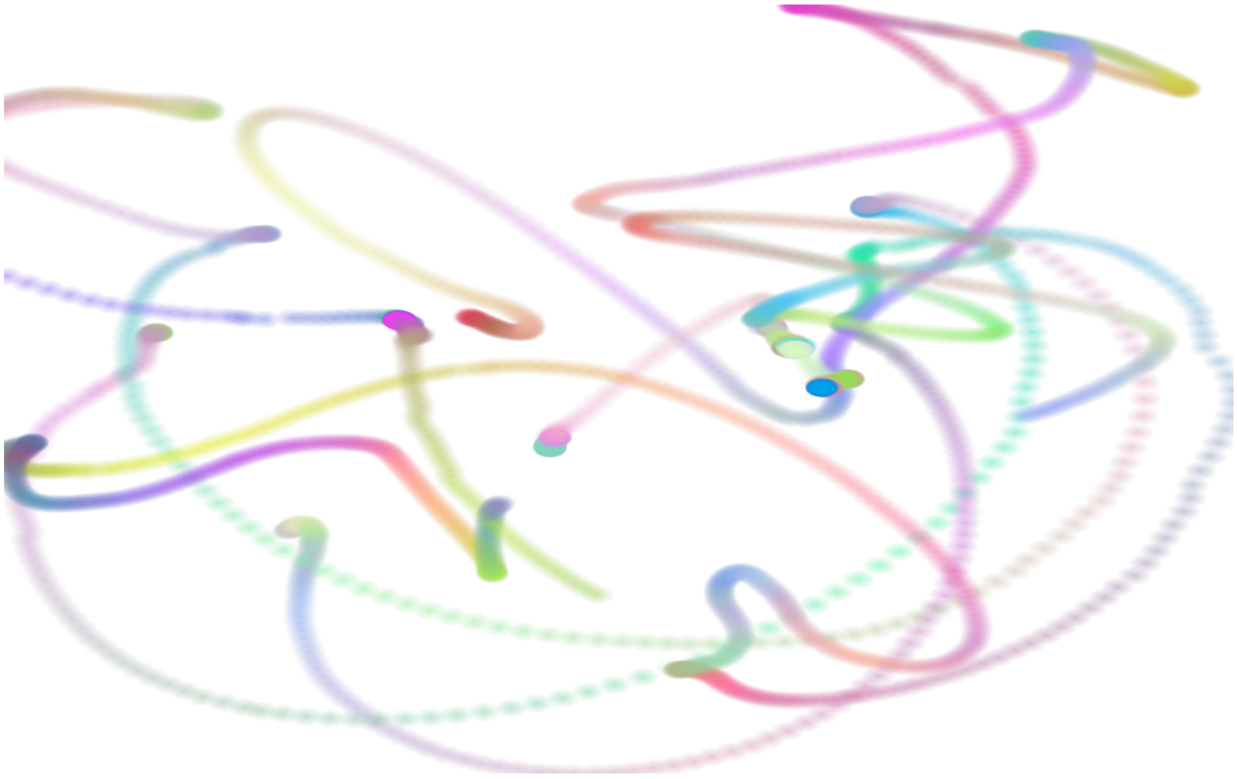
Pointillism with Color Variance



Airbrush



Airbrush with Color Variance



More images from different motions are located in the MoCapResults folder of my project! Check them out!

All motions are from CMU MoCap database subject #05.

6. FUTURE DIRECTIONS

Given indefinite time and resources, this project could go in innumerable directions. Some of the most interesting ones are:

RealTime Engine:
Because so much of the concept behind this project was based on a fascination with realtime interaction and how feedback influences our artistic decisions, succeeding in connecting to the RealTime Engine, requesting, and recieving data

User Interface:
-The current setup is not user friendly, only end-result friendly. I would like to add a small GUI to select options before painting (whether painting from pre-captured data or in realtime). However, while painting, only the "canvas" will be visible for artistic coherence.

User Study
Naturally following from the Real Time and UI are user comments on the program and their likelihood to use the program for different reasons. I would sample from dancers, choreographers, actors, and non-performers to get an idea of other possible applications of similar programs for other audiences as well as the success of this program for different audiences.

Viewing Planes:
-I would love to add alternate viewing/mapping planes. Mostly, I would love to see a 3D painting from the captured motion. The difficulty here is

Brushes:
-Use input points as control points for curves (B-spline, Bezier). The main difficulty here is knowing when to start and stop a curve because there is no LbuttonDown call when motion capturing.
-Painting with different colors or brushes with different end effectors.
-Adding an oilpaint brush that uses dynamic sizing during one stroke and also shades to capture light on paint.

AMC/ASF files
-Although parsing MoCap files was not what I had in mind, having parsed ASCII txts of C3D files, I think it could be helpful to also accept ASF/AMC pairs or BVH files (which are all ASCII files), extract the motion data, and create a painting. This way, motions captured on other systems can be used later to reflect on the choreography or performance.

7. CONTRIBUTIONS

Vicon: learning Blade, learning about connecting to servers, learning about realtime capabilities, learning about marker sets and skeletons,

Virtual Painting:

Learned GLUT, learned more OpenGL

ASCII txt of C3D file parser
Procedurally generated Virtual painting

8. REFERENCES

Guides:
Vicon RealTimeSDK Manual
http://bdml.stanford.edu/twiki/pub/Haptics/MotionDisplayKAUST/RealTimeSDKmanual.pdf

Vicon RealTime Data
http://grouplab.cpsc.ucalgary.ca/cookbook/index.php/Toolkits/ViconRealTimeData

CMU Database: http://mocap.cs.cmu.edu/

Info on C3D files: C3D.org

Blade Help Files

MoCap file interpretation: http://www.tabinda.net/mocapsim/software2.htm

Academic Papers:
[1] F. Crow and C. Csuri, "Music and Dance Join a Fine Artist and a Paint Machine," IEEE Computer Graphics and Application, pp. 11-13, 1985.

[2] El-Nasr, Magy Seif and Thanos Vasilakos. “DigitalBeing: an Ambient Intelligent Dance Space,”

[3] Meador, W. Scott, Timothy J. Rogers, Kevin O’Neal, Eric Kurt, and Carol Cunningham. “Mixing Dance Realities: Collaborative Development of Live-Motion Capture In a Performing Arts Environment.” ACM Computers in Entertainment, Volume 2, Number 2, April 2004.

[4] Baxter, Bill, Vincent Scheib, Ming C. Lin, and Dinesh Manocha. DAB: Interactive Haptic Painting with 3D Virtual Brushes.

[5] Fischer, Jan, Dirk Bartz, Wolfgang Straßer. Artistic Reality: Fast Brush Stroke Stylization for Augmented Reality.

[6] Hertzmann, Aaron. Painterly Rendering with Curved Brush Strokes of Multiple

Sizes.

[7] Henzen, Alex, Neculai Ailenei, Fabian DiFiore, Frank Van Reeth, John Patterson. Sketching with a Low-latency Electronic Ink Drawing Tablet.

9. Work Log

Week 1 (Jan 25, 2009 – Jan 31, 2009) – This week involved meeting with Joe and Alla and getting some feedback. The most significant part I had not really considered was file format (I considered physical data, eg would I use position or joint angles/skeleton or something else?). Both Alla and Joe directed me to CMU's mocap database and resources/tools available there. I also realized I really ought to do a formal user study to get feedback for final tweaks and future plans for the project.
Finally, I continued amassing and skimming through papers related to dance-MoCap, to follow-up on shortcomings of my original design document.

Research Summary
January 22, 2009

Project Abstract

Over the last couple decades, many dancers, graphics artists, and computer scientists have explored some of the potential applications of computer graphics and animation to the creation and performance of dance. Body Paint is a real-time interactive motion-capture program where a dancer will get immediate "feedback" on her movements in the form of a 2-dimensional virtual "painting." The dancer has a variety of options: the quality of the stroke (oil paints, charcoals, or ink, etc.), the view of the "painting," and the way the mapping is generated. Other factors, such as color and weight of the stroke, will be interpreted by the algorithm and based solely on her movement. When performing with Body Paint, not only will the dancer's movement influence the developing image, but the image will in turn influence how the dancer continues moving and uses the performance space.

Week 2 (Feb 1, 2009 – Feb 7, 2009) – I began collecting papers regarding brushstrokes and realized how overwhelmingly vast a topic that is. Many seemed far more complicated than what I needed, and others working toward such an unrelated goal as to be irrelevant.

Week 3 (Feb 8, 2009 – Feb 14, 2009) – I spent some time looking through the CMU database and resources. I conveniently found they have at least one subject with 10 or so captured dance motions.

Update on outline and some new details
February 8, 2009

Because my abstract was not super detailed, here's the intro for your perusal:

Body Paint will interactively create a virtual painting based on the physical properties (path, velocity, acceleration, and angular velocity) of the movements of the dancer. The default process will do a "conventional" painting – movements will map to a "canvas" at the front of the space. Ideally, there will be two other mappings: a painted "forest" where the motions will be traced and marked in place, with strokes applied, and a moveable cross-section will be projected; and a footwork painting, which only paints from the motion of the footsteps on the ground. These strokes will be selected at the beginning of the piece from a small group: oil paintbrush, ink/calligraphy, and charcoal/pastel. These may also be changed in the resulting painting after the fact. The color of the medium will be interpreted based solely on these properties, although the spectrum can be limited t the dancer's specifications.

Now, some elaboration and new knowledge:

To begin with, a little while ago, I met with Joe and Alla to go over my design doc and my basic idea. Some things I did not know/have reconsidered/adjusted:

Motion Data

CMU has an exquisite database of captured motions, including, dance motions! (http://mocap.cs.cmu.edu/ – note subject #5) This will be a great resource when testing out strokes and mappings. It has also been helpful (and will definitely continue being helpful) for figuring out the Vicon system and its methods.

Data Format

Before I can really get to mapping the movement to a path and applying a stroke, I need to think about file types. I've never worked with recorded mocap data, just real-time, so I didn't think too deeply about how the data was stored. I knew the system located a series of hyper-reflective balls in 3D space. I now know this data is then translated into joint angles for most later levels of processing. I am currently leaning to the raw-er data of c3d, with simply the positions of the markers. However, I need to figure out how to read the data in any file type used. CMU has an .asf/.amc reader. The software "ViconIQ" that comes with the system processes the camera data and ultimately outputs a .vsk/.v. I am unsure as of now in what data formats Vicon captures/processes/records in real time and which one is ideal.

User Interface

First and foremost, I should begin planning this out presently, not toward the end was originally prescribed. To that end:

I do not want it to use gesture recognition. Gesture recognition requires new learning that is orthogonal to the actual goal (creating a virtual painting by dancing). Instead, I would like a conventional and usable interface on the home machine of the software, something simple and intuitive for the accepted, learned interaction with GUIs. It will be a short series of choice:

* What stroke would you like to use?: Oil paint, calligraphic ink, charcoal or pastels

* What color range? Select range of hue, saturation, and value
* What kind of mapping would you like to use? Conventional canvas(movements are interpretted as strokes on a virtual canvas at the “front” of the 3D dance- space, regardless of “depth”), 3D painting (paths follow the movement in 3D space with a moveable viewing plane that can cut through the space anywhere), or footwork (only motion that contacts the floor will leave a path, resulting in something more like pointilism)

These will be a conventional, simple interface on the base machine. Once the dancer has selected her choices, she will get in place to begin her piece, and the technician will select an option to start recording.

After completing the piece, I would like the dancer to be able to change the stroke or color selection to see other resulting paintings.

Notes:

-Joe recommended using prerecorded bvh files (such as those available from CMU) to show off the various strokes in the interface.

-If I get different mappings working as intended, I want to demo the resulting “painting” in the interface – ie, a 2D painting, a 3D painting, or a “floor” painting from the same dance.

User Study

-In my head, I assumed I would ask people to come in and try it out and give feedback. I should clearly formalize this and include it in my deliverables. This is a project about user experience, after all.

QUESTIONS:

Brushstroke Algorithms:

-I am having some trouble finding useful papers for simple implementation of stroke algorithms. I find all these papers on painterly rendering which seems more complex than required, and some on different interfaces or applications, but none simply on taking a path and applying an artistic stroke, a la Photoshop and Illustrator. I remember some things on defining curves from Intro Graphics which I could probably use to smooth the strokes, but regardless, I need algorithms for the strokes so I can get onto implementing them. I’d appreciate any help parsing through the vast literature on this and related topics to narrow down to very relevant topics. I will keep looking, but I feel like these are pretty standard, I am just not exposed to them.

I am a bit behind, but I have reworked my schedule to optimize catching up and getting back on track.

Posted in Overview | 3 Comments

Week 4 (Feb 15, 2009 – Feb 21, 2009) – Friday Feb 20 was Vicon training day where I learned what is different about our system from the one I used and got a refresher on calibrating and capturing. I learned we do not currently have a set suit, which I would prefer. However, as an ex-dancer, I own a surprising amount of spandex that can be used. I also have access to the PACshop which, I believe, may have some lycra suits.

Week 5 (Feb 22, 2009 – Feb 28, 2009) – Due to fairly unfruitful initial searches into how on earth to write these brushstrokes in efficient and simple algorithms (just the basis, not the coloring or variation), I continued my literature search and decided that I am simply not going to find as clear a reference as I was hoping. So it seems when I get back, I will be thrusting myself blindly head-first into choosing a windowing system and applying stroke algorithms to mouse data.

Week 6 (Mar 1, 2009 – Mar 6, 2009) – Compiling the Alpha review. Takeaway from Alpha meeting with Joe: working with brushstrokes, printing out coordinates in realtime, meet with Alla after spring break as soon as possible.

Weeks 7&8 (Mar 16, 2009 – Mar 28, 2009) – Tried to arrange Alpha meeting with Alla, however she got sick and we had to postpone until Mar 31. I set up the coding environment for brushstrokes, but it's buggy.

Week 9 (Mar 29, 2009 – Apr 4, 2009) – Finally had Alpha meeting with Alla on Mar 31.
MoCap realtime print out-does it need to be on a different machine?
-how to connect from same machine?
Adding on top of buffer instead of refreshing?
Start with dots and brushstrokes
Then working from c3d data
Find c3d parser!!

Tried capturing data and configuring/calibrating it with Blade. Very very bad auto-solving and incredibly flickery points.

Week 10 (Apr 5, 2009 – Apr 11, 2009) –

Fri, Apr 10, 2009 at 4:51 PM Got the videos from Joe on ViconBlade. Realized that the data is streamed directly out by the RTE and worrying about file types like c3d or bvh is an unnecessary processing step which will interfere with runtime latency.

Finally debugged my GLUT errors. I was having issues drawing anything to the screen. Converted coordinates from mouse to view.

Week 11 (Apr 12, 2009 – Apr 18, 2009) -

Tue, Apr 14, 2009 at 1:58 PM

Also, I tried to watch some of the movies at home over the weekend but, evidently, I lack a plugin but quicktime never specifies which plugin so i didnt know what to install and figured i will just try to watch in the lab this week.

-Spent a bunch of time figuring out Sockets and Ports and connecting to servers. Watched a lot of Blade videos, and multiple times.

Week 12 (Apr 19, 2009 – Apr 25, 2009)
Sun, Apr 19, 2009 at 5:12 PM

It turned out not quite as successfully as I would have liked (I would have emailed you but I have literally been going straight since then.) The rest of thursday, I could not get a skeleton to load to my points (I don't know Nexus as well) and Michael was there wanting to film some MoCap for those TVs, and I only wanted to hold him up so long, so I had him film just the markers and me dancing around, and then I had a meeting to go to so I closed everything down. When I went back on Friday, I tried playing with Blade and the RTE and getting the code to run and looking up more things about that and did not make any legit progress in the time I had. I tried to find stuff on socket errors and stuff on running Vicon in real time. The only thing I may have found was that I might need to make a skeleton or object for what I want to do in order to run the RTE. I also tried to do this (make a skeleton or object) and though I found where you can make one, I couldnt figure out how to specify stuff about it (again, didnt have loads of time). So this means:

1) Trying to figure out what's giving us that Socket error in the sample client code

2) Watch more of the videos on RTE and, evidently, making skeletons/objects.

3) Make a skeleton or object for my code

4) Get the real time thing working and pumping out the x y zs like it should and hook it up to some brushstroke code.

5) Rework my brushstroke code because it's a bit convoluted right now.

6) Make brushstrokes prettier.

7) Play with color and weight.

The last 3 were more for myself than for you. I was kind of on a roll... anywho, I will be there tomorrow at 5. Hopefully you are there in case you have any good ideas for stuff, but if not, it's a good environment for me to work on the brush stuff. I know, normal people leave at 5. Hopefully we'll overlap?

Tue, Apr 21, 2009 at 7:22 PM
I got some of it working! I can explain more later, but the long and short is that I did get a blade skeleton to map to the markers. I saved the range of motion capture that mapped, if weakly, to a skeleton. Main problems:

-I don't know what the marker set actually is for this skeleton.
-I forgot to wear the hat and so have no head markers (not that I care, but the skeleton does)
-I try to get the RTE to work like it says to for motion builder, but when i replay last capture, nothing plays in the perspective window. I also noticed that noting would show up in the perspective window except after processing the ROM/associating them with a skeleton. Ideas?
-I actually can't figure out how to relabel mislabelled markers. The main solve is good but, because of missing the head and not knowing the marker set, it's a bit wonky. But it does work. Significantly less flickering.

Oh, and once I did the supposed steps to connect to the RTE (which didnt look like it worked in Blade, anyway), I tried running the example code and got the same socket error: 10061:

Connection refused.

No connection could be made because the target computer actively refused it. This usually results from trying to connect to a service that is inactive on the foreign host—that is, one with no server application running.

- I was considering calling Vicon about all the issues with Blade. However, as I was writing up a list and reviewing my resources for troubleshooting, I had a couple ideas about how to solve some of the issues: playing with thresholds. Rather than be told this over the phone, I set to work. Most significantly, increasing the strobe intensity of the cameras improved the capturing of markers in Blade, but many markers were still lost. However, the slight improvements came at regular intervals and, though minor, gave me a better understanding of the software as a whole.

-The bad connection was an incorrect port. After reading a lot of vague literature on the RTE (which, supposedly, always connects to port 800) and ip addresses and open streaming ports, I finally discovered that our RTE is on port 801. Significant!

Week 13 (Apr 26, 2009 – May 1, 2009) - Finetuning and debugging brushstrokes, creating presentation, and trying like hell to get ExampleClient successfully connecting and streaming data. No dice. Added color variance and got the Ink brush finally working, which varies radius with instantaneous velocity.

Week 14 (May 3, 2009 – May 8, 2009) - Looking up bvh v c3d v amc/asf

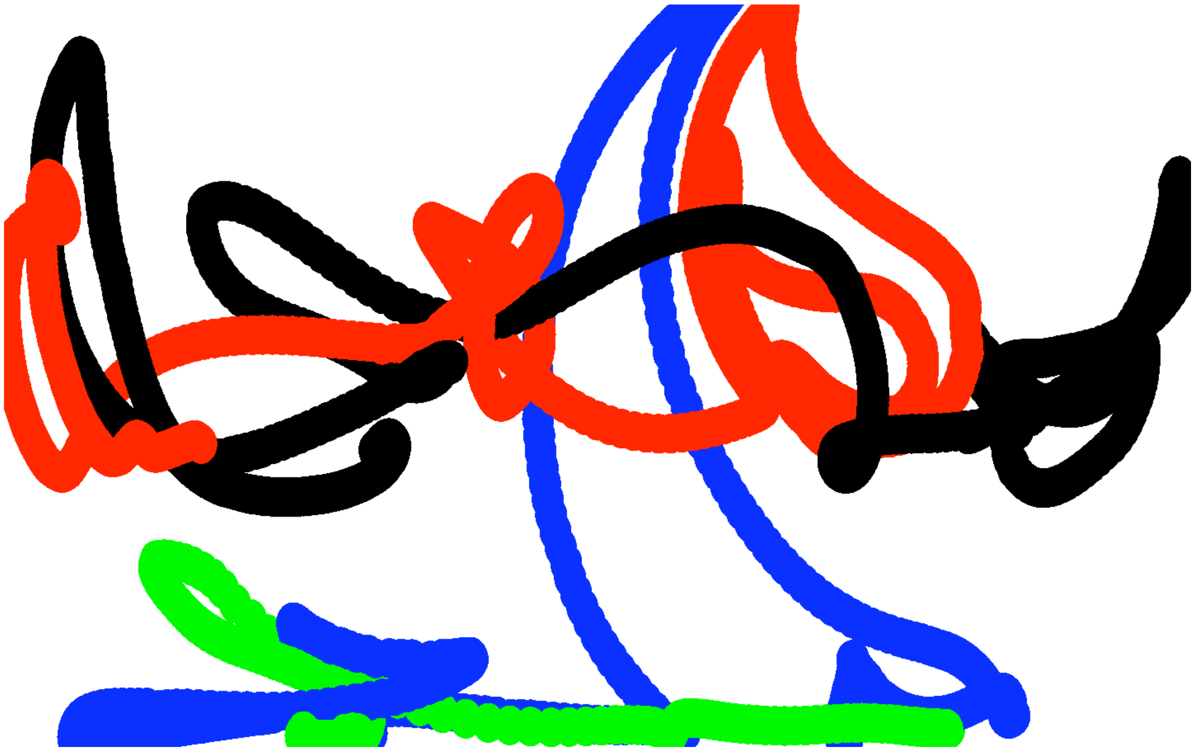
: I went with c3d (technically, an ascii txt converted from c3d). I actually nixed bvh and thought I was going to do asf/amc file pairing, but that data management got more and more complex, and so I figured, since I was having more issues parsing than I thought (I thought it'd be more straightforward, but I got held up in the tokenizing phase) and, therefore, need data that, though vague, is in the form I need. This will all be in the paper, but: the reason I first tried asf/amc is that the only c3d converter I could find for free does not label markers and, as I mentioned, c3d files add 100-300 markers to a standard marker set. Even when these are labeled in Blade, this c3d-ascii converter just labels them 1-numMarkers. I did find a trial of one other c3d converter, however, it is probably even more unreadable (by this I mean, harder to find salient data amidst the file) and I can only convert files of <=100 frames. So yeah. asf/amc are beautiful files to look at, but unwieldy to parse. I was ambitious and thought, figure out the basic unlabeled c3d, then try asf/amc and do some pretty math that I need to learn and tada, fancy project. Not so.

Week 15 (May 9, 2009 – May 13, 2009) Finally finished the ASCII txt c3d parser! It only works on CMU's marker set currently, but it successfully interprets those files! The code parses the file into frames, extracts the lines with the desired markers (4/330, it's a lot of data for little salient info), and tokenizes. Takes the x and y values f all four markers, stores them for that frame, and moves on to the next frame. Once you hit 'g' it draws under the current specifications.

I made one final change. I found the paintings from the x and z axes (z is the up vector) incredibly boring compared to the x and y plane. Perhaps it is because of the limits of the short motions, but these paintings generally had two areas of brushstroke – one toward the top and one toward the bottom, with a chunk missing in the middle. This is because these motions do not have many level changes – one kick, for example. These paintings look more like medical graphs than paintings. So, instead of yielding a vertical frame at the front of the space, the program currently provides a top-view painting – essentially, a trace of where you've been.

Pictures are included below.

Front view of 05_10:



Top view of 05_10: